

Validator

for Identity Manager

User Manual

last updated: 5 Sep 2014

Table of Contents

[Table of Contents](#)

[General](#)

[Change Log](#)

[1.3](#)

[General Q&A](#)

[Tips and Tricks](#)

[Inline help](#)

[Variables](#)

[Connection settings](#)

[Testing Connections](#)

[eDirectory Connector](#)

[Active Directory Connector](#)

[Text File Connector](#)

[Attribute names](#)

[Test Object Naming](#)

[Unit tests vs Integration Testing](#)

[Avoid Pause actions](#)

[Setup/Test/Cleanup](#)

[Setup](#)

[Test](#)

[Cleanup](#)

[Run a Test Suite via the commandline or cron](#)

[To execute a test suite from the command line, run the script with the test suite file as a parameter.](#)

[Known Issues](#)

[Connector FAQ](#)

[LDAP Based Connectors](#)

[AD Connector](#)

[eDirectory Connector](#)

[Execute Connector](#)

[HTTP Connector](#)

[User App Starting a Workflow via SOAP](#)

[JDBC Connector](#)

[Text File Connector](#)

[User App Connector](#)

[Sample Error Messages:](#)

[General Connector errors:](#)

[Bad DN in a FDN field \(note the plus sign\):](#)

[Typo in attribute name during object creation:](#)

[Typo in attribute name in a modify event:](#)

[eDirectory Connector Specific:](#)

[Attribute in object class not yet on user:](#)

[Single valued attribute in eDirectory, trying to add a second value:](#)

[Previous Change Logs](#)

[1.2](#)

[1.1](#)

[1.0](#)

General

Current available version: 1.3

Installation

- Unzip the Validator zip file
- Start the Validator server (in the unzipped folder)
 - Windows: runValidator.bat
 - Linux: runValidator.sh
 - Mac: runValidator.command
- To start the user interface, access the URL: <http://localhost:1077/validator>
 - Note:
 - Tests are loaded from %INSTALL_DIR%\tests (or /tests on Linux)
 - Log files are stored in %INSTALL_DIR%\logs (or /logs on Linux)
- To execute a test suite from the command line, run the script with the test suite file as a parameter.
 - Example: runValidator.sh "tests/myTest.json"

Licensing

Validator comes with a 30 day license which starts counting down after the Validator server is started for the first time.

For questions regarding pricing and licensing, contact your sales representative.

To install a license:

- Replace the %INSTALL_DIR%\config\license.dat file with the new license file (or config/license.dat on Linux)
- Restart the Validator server

Upgrades

Unpacking a new version of the Validator archive file over the top of a current Validator installation will overwrite any changes made to files in the config folder. Files you may have updated in the config folder are:

- validator.properties
- license.dat

Before unpacking the new version, please backup any files you may have changed in the config folder, then re-apply those changes after installing the new version of Validator.

Change Log

1.3

Enhancements

- Runner <http://localhost:1077/runner>
 - Run all test suites from one location and consolidate the reports.
- Reporting can now be in RTF format.
- Implemented test grouping to allow for easier management of test groups.
- Results tab will auto-refresh by default, and will stop refreshing when test execution has completed.
- Implemented new Manual Test action in the Generic Actions connector. It is similar to Pause but it lets you pass or fail the test.
- Import IDM Driver connection information from NetIQ IDM server
- Added button to convert the current connection fields to variables
- Renamed variable groups to “Environment” to help clarify their purpose.
- Added a Cleanup icon in variable management to remove all unreferenced variables
- UserApp connector - added directory browse buttons to specific fields
- Implemented new time generic action to manipulate time formats
- Implemented new Postgres connector

Bugs

- Couldn't drag an action into an empty category in the Test Editor
- Hide LDAP filter message in LDAP browser if the filter was previously being used
- 880623 Implement HTTPS in HTTP connector
- 880588 HTTP Connector GetPostPutIntoVar is not working
- 883637 UserApp: Include hover over information for fields.
- 883638 Enhancement: Change the field names to match required data.
- 848733 Report will not run on empty test
- 850703 UserApp connector - support multi-value fields on request form

- 871425 CP: Unsupported Browser did not popup for IE
- 781735 Refactor existing LDAP move call
- 883292 IDM Validator does not correctly assert values not present when no values left in attribute
- 735706 Validator needs ability to import connection/path/etc. information from vault.
- 783072 The AD DNs can contain a CN with a comma and Validator does not handle this
- 874849 Manual Test needs to work with the commandline
- 885524 UserApp: Browse tree for object fields
- 874854 CP: Upgrade jquery-ui theme
- 871666 Add grouping functionality
- 885538 Remove unused header fields from reporting
- 872145 CP: change Jasper to do rtf, pdf
- 874819 CP: Dress up the pass/fail on the manual test. Also include pause refactor
- 769675 LDAP browser needs to support any number of objects in a container
- 814325 Results tab/log viewer auto-reload defaults
- 849108 CP: if json test file is newer than Validator engine, then an error message should be returned
- 872155 CP: pure manual test
- 881374 Refactor: Fixed the boolean strings to be true booleans throughout the project.
- 685631 Include 'base' driver templates
- 696846 Add some more details to the view log so it's easier to see.
- 849105 should tell you which version of the json file is loaded
- 781513 Be able to type in DN of imported user or use the selected user to import attrs
- 813221 Error: The JsonSerializer LongDeserializer failed to deserialized json object "" given the type long
- 760765 Need to be able to update a value from a multi values attribute
- 784311 Show all text of a Pause Action
- 817873 Support variable expansion in Retry Count/Interval fields
- 884094 Initial Validator load prompts for Test Suite File Name with empty dialog
- 818013 Get Object FDN and Set Variable doesn't do anything when object not found
- 818016 "Get Object FDN and Set Variable" doesn't handle \ in the FDN value
- 850691 command line tests don't execute templates
- 850693 command line tests don't calculate the current value of variable
- 811545 Not able to add character '\' used in distinguishedName of AD in variable
- 812817 Comma in CN of LDAP DN is Broken
- 653116 Drag and Drop between pretest, test and post test
- 874640 Create and manipulate timestamps
- 880773 XPATH action in the generic actions connector
- 669599 Upon deleting attribute something should flash next to the X to show it happened
- 707714 Intruder Lockout triggering
- 850705 Get date and do date arithmetic

Change logs for other versions are at the bottom of the document.

General Q&A

Q) Where can I find some sample tests?

A) <https://groups.google.com/forum/?fromgroups=#!topic/novell-validator/Dxmz05Tiflk>

Q) I need some new feature? Or some feature does not quite work right?

A) Submit a bug in [Bugzilla](#). To get priority, consider discussing the size of your environment. A customer with a million user license has more weight than a customer with a 10 user license, alas. In case you forget, you can always click on the question mark icon (?) in the upper right hand corner and the starting page of Validator is shown, which has a link to [Bugzilla](#) available.

Q) Is there a support forum? A community of users?

There is a [Google Group](#) available to try and get support from others like yourself. You can find a link on the Validator startup page, which you can get to, by clicking on the question mark (?) in the upper right hand side of the screen.

Q) There are no OK buttons in the forms so I can save or apply changes.

A) All changes you make in the UI are immediately applied to the Test Suite document in memory. You just have to click on the Disk icon when you want to save the document to the server (the default location is the “tests” folder where the Validator is installed). This may make you nervous at first, but you'll get used to it, and it saves lots of clicks in the end.

Q) How do I get back to the main page that is shown when the Validator first starts in the browser?

A) Click on the ? icon in the upper right corner. From this page you can click on links to enter a bug against the Validator for IDM or read this FAQ.

Q) How can I reorder tests on the left side?

A) You can drag and drop both Tests and Connections so they are in the order you wish.

Q) Firefox on some SLES versions does not work correctly.

A) Firefox 3.6 or greater is required. TIP: If a VM has an old version of Firefox, just use Firefox on your host to connect to Validator for IDM server running on the VM. (See the URL at the top of this document.)

Q) Can I change the directory where test suite files reside?

A) Edit **config/validator.properties** and specify the desired folder for the TESTS_LOC property.

When specifying Windows folders you must either use slashes or double-backslashes.

For example: c:/dir1/dir2/mytests or c:\\dir1\\dir2\\mytests

Q) Can I change the default values for Retry Interval and Retry Count that are in every assert action?

A) Carefully edit the config/Validator_Schema.json file and search for Retry Count, then change the default values. It would be wise to make a backup of the file before editing it.

Q) What is the best way to monitor a running test?

A) The Results tab will show a live view of the current log file if you click on the auto-refresh icon. If however you want to view the log from the OS, all logs are in the validator\logs folder..

Q) How do I allow outside connections to Validator?

Validator 1.3 and 1.3.1 allows all outside connections.

Validator 1.3.2 provides the ability to limit connections based on configuration in validator.properties.

Edit file:

%install_dir%\config\validator.properties

The property value is a regex expression that allows specific ip address patterns, separated by the pipe (|) character. By default it restricts access to localhost only by allowing all ip addresses with start with 0 (for IPv6) and 127 (IPv4).

Default value:

REMOTE_ADDR_FILTER=0.*|127.*

To listen on all IP's, change the line to:

REMOTE_ADDR_FILTER=.*

To add a single outside IP address to the default list, simply list it as follows:

REMOTE_ADDR_FILTER=0.*|127.*|172.17.2.99

Validator server needs to be restarted after modifying the property file.

Q) How do I change the Validator port from the default 1077?

Edit file:

%install_dir%\config\validator.properties

Modify the MAIN_URL property to reflect the desired port.

Restart the Validator server.

Tips and Tricks

Inline help

If you are not sure what a field takes or does, mouse over it, often there is a tool tip with useful information if you are patient for a second or three.

Variables

Version 1.1 introduced variables to Validator. Variables can act as constants, much like IDM GCV's, and they can also be assigned new values during test execution. In order to enforce the disciplined use of Variables and eliminate the side-effects of mistyping variable names and accidentally overwriting values, Validator requires that all variables be declared before using them.

Declaring variables can be done in two places; the Test Suite tab -> Variables menu, and in the variable browser (V in a scroll icon that appears in various areas



of Validator). Variables do not need an initial value if you are simply declaring them so they can be assigned values within a test.

Referencing variables can be done in any text field in Validator. Most places you may want to insert variables will have a (V) icon somewhere on the page. This icon brings up a variable browser listing all declared variables. If you want to declare a variable in the browser, click the + sign icon. When you select a variable to use, it will insert a variable reference in the following format: \${variable name} in the LAST TEXT FIELD you were editing. If there is not a (V) icon on the page where you want to insert a variable reference, you can do it manually by typing in the same variable reference that the variable browser inserts. If you reference a variable that has not been declared or has been deleted, Validator will treat the reference as literal text. If you attempt to assign a value to an undeclared variable (because the variable was deleted at one point), a pop-up will be shown during test execution indicating that the variable doesn't exist. To fix this, simply declare the variable.

Dynamically assigning values to variables is done in connectors that support such actions. Actions that assign values to variables are found in the LDAP, JDBC and Generic Actions connectors. More will be added as needed. Actions typically have a description of "set variable" in their names. See the Connector FAQ for details on these actions. For example, the LDAP connector has an option "Get Object FDN and Set Variable" to place the object's DN into a variable.

Validator 1.2 introduced variable groups. Validator 1.3 renamed them to Environments since Variable Groups are used to switch between different testing environments. You can have a set of variables for a Development environment, and one for a QA environment, etc. As long as you use variables in your connection definitions and in your tests, you can easily switch environments. See the help page on the Manage Variables page for more information.

Connection settings

In general you should consider using multiple test suite files for different testing tasks.

However, if you are able, define all your connections once in a single file, and then use a copy of that file as a base to start building all other test suites. This maintains the original GUID's for the various connections which is how the system internally identifies them. Thus it would be possible to later copy a test from one suite to another, and not have to fix the GUID references back to the connections.

Testing Connections

- When you define a connection, select it, and the 'Play' button will let you test it. If it fails, look at the Results tab for more detailed error

messages.

eDirectory Connector

- The DN of the bind user should be in LDAP format. cn=admin, ou=system, o=sa
- If you switch from clear text to SSL via the check box, also change the Port to 636 it does not do it for you, as some tools are wont to do.

Active Directory Connector

- The DN of the bind user should be in LDAP format. cn=Administrator, cn=Users, dc=domain, dc=com (Remember that the default Users container in Active Directory is cn=Users, not ou=Users as you might otherwise assume).

Text File Connector

- The files have to be local to the server where Validator is installed (though you could point at a network share for the file).
- File name vs File Pattern: Only select one. Either an explicit single file. Or else a Regular Expression pattern in File Pattern and all files that match that pattern will be considered.

Attribute names

Generally attribute names for Active Directory, eDirectory, and LDAP Connectors are in the LDAP name space. Specifically for the eDirectory connector, it would be loginDisabled not Login Disabled. Use an LDAP browsing tool to learn the LDAP names if required. Be aware that each eDirectory server has a attribute map that allows the administrator to remap the name LDAP shows for any specific attribute. Thus using a real LDAP browser is the easiest way to be ascertain the correct name of the attribute.

Test Object Naming

Consider developing a naming pattern for the objects your tests generate. This will make it easier to track and deconflict more than one tester or test suite at a time in your identity system.

Unit tests vs Integration Testing

In general, consider having different test suites for different use cases. Unit testing of an IDM drivers functionality is a different use case than integration testing for the end to end solution.

In the Unit test you want to test small pieces of functionality, (small is ill defined in this context, depends on the circumstances). In Integration testing, you do not need to retest all the unit tests, rather present the start conditions, and test for the resultant end conditions.

Avoid Pause actions

One of the interesting things about Identity Management is that it can take time for events to process. Particularly when there are many connected systems involved, or perhaps events have to flow through a series of drivers.

In some environments, certain tests can take longer than expected in the best case. Thus you might be tempted to use Pause or Delay actions to help add some delays for event processing.

However, consider instead asserting that some known attribute exists, since each such action has a Retry Count and Retry Interval setting. These actions will complete, once the result is found.

Thus to add a one minute delay you could add an appropriate Assert Attribute or Value exists, and then set the retry count to 10, and the retry internal to 6000 (milliseconds) and then it will try until it succeeds (thus taking the minimum amount of time required) and continue trying for a full minute. This way your tests will likely run faster than simple hard coded pauses.

Setup/Test/Cleanup

Each test contains three sections.

Setup

The basic notion is that first, Setup the environment and make sure that everything looks as it needs too in order for the tests to work. This would include things like deleting the objects used for this test, and perhaps Assert Object does not exist to be certain that the delete succeeded. The cleanup section normally should do this, however a test might fail, and not get to the Cleanup section. Thus as setup to each test, make sure to clean everything that might be left behind.

Test

In this section, first create any objects needed to test with. Then generate the event being tested for. (Like changing a password, moving an object, changing a flag attribute, etc). Then test for the expected results.

Cleanup

Finally, once the test succeeds, cleanup any objects related to the test and reset the directory back as close to starting conditions as possible. Do not forget to clean up in all connected systems that might be affected, not just the identity vault.

Run a Test Suite via the commandline or cron

To execute a test suite from the command line, run the script with the test suite file as a parameter.

Example: runValidator.sh "tests/myTest.json"

The results will be in the corresponding log file in the logs directory.

Known Issues

P) Internet Explorer doesn’t seem to work right.

S) What can I say... IE is a troubled child. Use another supported browser specified on the main page.

P) I have copied my Oracle jdbc driver to the lib/ext folder, and restarted the server, but I still get a ClassNotFoundException trying to load the Oracle jdbc driver

S) In some cases, java libraries (like Oracle's jdbc) are packaged in .zip files instead of .jar files. If the Oracle jdbc driver is packaged in a .zip file, try renaming the file to have a .jar extension, and restart the server

P) Using the MSSQL connector displays an error: ClassNotFoundException: net.sourceforge.jtds.jdbc.Driver

S) Validator v1.1 ships with the JTDS JDBC driver, however it is not in the classpath so the driver cannot be found. To fix this, edit the runValidator batch or script file and include “lib/mssql/*” into the classpath. Then restart Validator.

P) When attempting to run a test, the following error is returned: Failed parsing JSON source: java.io.StringReader

S) These problems were resolved in v1.2. If you have a testsuite with this problem, contact support and send the test suite json file.

P) If an assert retries for more than 5 minutes, the server technology used by Validator will automatically restart the test. This will cause two threads to execute the same test so you will see it continue to retry in the original thread, and the new thread will start the test over.

S) This is resolved in 1.3. In earlier versions to prevent these retries from occurring, modify the Validator startup script with the following:

`-Dsun.net.httpserver.idleInterval="3600"`

This parameter instructs the Jersey http server to allow 60 minutes of retries, which should be sufficient for any Validator test.

Then restart the Validator server.

Use the following example to see where this parameter should be placed.

```
:runUIMode
start jre/bin/java.exe -cp
lib/*;lib/ext/*;lib/enc/*;lib/jldap/*;lib/json/*;lib/junit/*;lib/mysql/*;lib/rest/*;lib/ssl/*;lib/oracle10g/*
-Dsun.net.httpserver.idleInterval="3600" com.novell.nccd.validator.RESTServer $1
GOTO :EOF
```

P) When executing runValidator.sh on some SLES servers the following error is returned:

Starting the REST Server

/libjvm.so'.oot/Validator/validator/jre/lib/i386/server

Please install or use the JRE or JDK that contains these missing components.

S) The issue is environmental in nature but can resolved with a simple modification to the startup script.

Edit: runValidator.sh

Following the chmod line, insert the following:

dos2unix jre/lib/i386/jvm.cfg

P) When executing testsuites from the command-line on Linux, you may encounter an error that the testsuite file could not be found. This may occur if the filename has a space in the name.

S) This is resolved in 1.3. In earlier versions edit runValidator.sh and at the end of each execution line, surround \$1 with quotes. e.g. “\$1”

Connector FAQ

LDAP Based Connectors

Q) Can I retrieve a value from a directory and use that value later on in the test?

A) Yes. There are actions that set values in variables, and those variables can be referenced at any time during test execution. These actions are:

1. **Get Object FDN and Set Variable.** This action will search the tree from a specific Base DN for an object whose attributes have the values you specify. The variable is set to the object FDN. If the object is not found, the variable is simply assigned an empty value, but no error is returned. Note: Be sure to populate the Base DN field or the search may not work on some LDAP servers.
2. **Get Value and Set Variable.** This action will set the variable value to the *FIRST* value from the specified attribute.

Q) There is a new field for Classes in the Create Object action. Do I have to use that field or can I specify classes as in previous Validator versions, as values of the objectClass attribute?

A) The Classes field is optional if you specify your object classes in the Attributes and Values table. However, when you specify classes in the new Classes field, the schema browser will list only those attributes specific to the classes listed in the Classes field.

Q) What does the Import Fields from Object link do in the Create Object action?

A) If you have an existing object in the directory that is similar to the object you are creating in Validator, you can import all attributes from that object which will automatically populate the Classes and Attributes fields in the Create Object action.

Q) How do I set dates - such as passwordExpirationTime?

A) Version 1.3 introduced time calculation and conversion actions in the Generic Connector.

Use this answer as a general reference about LDAP date formats.

Use the format: YYYYMMDDHHMMSSZ e.g. Jan 15, 2011 at 1:05:23 PM is 20110115130523Z for zulu (UTC) time. To specify time for your own timezone, replace Z with UTC time offset in the form: +/-HHMM where HHMM represents the number of hours and minutes in relation to UTC time. e.g. 20110115130523-0500 is 5 hours behind UTC time which is Eastern Standard Time.

Q) I don't know the full CN of a user object. Can I use wildcards when deleting objects and checking if an object exists?

A) You can use wildcards in the CN value for Delete Objects and Assert Object [Not] Exists. For example: cn=testuser*,ou=myou,o=myo would match all users whose CN starts with "testuser". Only * is allowed as a wildcard.

AD Connector

Q) How does Assert Password behave? (A.K.A. Why does my assert password not work?)

A) Active Directory has two possible password attributes exposed via LDAP.

1. [userPassword](#) (as all LDAP systems should)
2. [unicodePwd](#)

Each AD tree can be configured as to the behavior of these attributes. Please read MSDN articles linked for each.

The Assert password method has been modified to try a bind as the user with the specified password to avoid this problem, and care should be taken to avoid multiple retries as bad password attempts could trigger intruder lockout.

To set the password, you need an SSL connection to AD. In addition, it is best to set the password using the Set Password action as opposed to setting the password during an object create.

Q) Why can't I set a password on my AD user using Create Object?

A) You need an SSL connection to AD to set the password. You also cannot set the password during user creation, instead use the Set Password action after the user has been created.

Q) When I create an object in AD it is in a disabled state. If I try to set userAccountControl during object creation, I get an error "WILL_NOT_PERFORM". How can I enable the AD account?

A) You must create the user in AD without a password and without setting userAccountControl. Once the user is created, use the Set Password action to set the password. This is critical because you cannot enable the account without a password being set on the user. Then use Replace Attributes and Values to set the appropriate value in "userAccountControl". Set it to a value of 512 for a normal enabled account. See <http://support.microsoft.com/kb/305144> for all possible values for userAccountControl.

Q) If my AD is set to require user to reset password on administrative password reset, what happens?

A) This will return an error LDAP 49, subcode 773 ([List of codes](#)) which will be reported in the log, and counts as a login failure, even though it technically indicates the password is correct, but login via LDAP is not allowed until the password is changed.

Q) What possible error codes might occur on a LDAP bind attempt that the connector can return?

A) There are several, and build 335 and higher will report the following error cases in the log.

- 525 - user not found
- 52e - invalid credentials
- 530 - not permitted to logon at this time
- 531 - workstation restriction
- 532 - password expired
- 533 - account disabled
- 568 - too many contexts ids (too many group memberships)
- 701 - account expired
- 773 - user must reset password

Some of these 'errors' are not technically errors. For example in Assert Password equals, while 525 and 52e are clearly indicators that the password is incorrect, 531 (workstation restriction), 532 (password expired), 533 (account disabled), 701 (account expired), and 773 (user must reset password) only return if the password is correct, and thus in the context of an Assert Password test, mean the password is actually correct.

Q) How do I create a group in AD.

A) Use the "create object" action

Enter these two classes:

objectclass=Top

objectclass=group

Enter these attributes:

cn:MyGroup

dn:cn=MyGroup,ou=grou,dc=whatever,dc=com

sAMAccountName:MyGroup

name:MyGroup

Q) When a DN has a comma in the CN, how can I specify the DN in Validator.

A) When specifying the DN, escape the comma with a backslash.

Example: cn=Jones\, Fred,ou=user,dc=company,dc=com

eDirectory Connector

Q) How should I handle testing structured attributes? For example DirXML-Associations, DirXML-EntitlementRef which are both Path syntax.

A) Path syntax has three components, and in this connector are exposed in LDAP format of:

volume#nameSpace#path

volume = DN value in LDAP syntax like cn=DriverName,cn=DriverSet,ou=OrgUnit,o=Org

nameSpace = 32 bit integer (usually 0-4 for DirXML-Associations or 0-1 for DirXML-EntitlementRef)

path = A case insensitive string ([Possible DirXML-Association values](#), DirXML-EntitlementRef will be an XML nodeset of varying content).

For DirXML-EntitlementRef:

Use an Assert attribute value contains, with a value:

cn=EntitlementName,cn=DriverName,cn=DriverSet,ou=OrgUnit,o=Org#1#

For DirXML-Associations:

Use an Assert attribute value contains, Regex enabled:

cn=DriverName,cn=DriverSet,ou=OrgUnit,o=Org#[0-9].#.+

where [0-9]. means 1 to many instances of a number for 0 to 9.

Q) How should I grant/revoke an entitlement and check for it?

A) We don't have an "entitlement" action yet but you can still do it by adding the correct attributes.

You'll want to tweak this for your specific entitlement but the steps are more or less:

- Make sure DirXML-EntitlementRecipient is added as an objectClass.

To check for an entitlement grant/revoke check the attribute "DirXML-EntitlementRef" for the appropriate value. For example:

This is a grant:

DirXML-EntitlementRef attribute equals "cn=TestEntitlement,cn=Loopback Driver,cn=driverset1,o=system#1#<ref><src>RBE</src><id>system\driverset1\Entitlement Policies\Test</id><param>TestValue</param></ref>"

This is a revoke (notice the #0# instead of #1#):

DirXML-EntitlementRef attribute equals

cn=TestEntitlement,cn=Loopback Driver,cn=driverset1,o=system#0#<ref><src>RBE</src><id>system\driverset1\Entitlement Policies\Test</id><param>TestValue</param></ref>

So you can have whatever creates the entitlement run and then check for it. i.e. workflow, adding role, etc.

If you want validator to artificially add an entitlement you do it the same way by adding an attribute with the specific value to revoke or add.

Execute Connector

Q) How do I run a script?

A) Define a connection, using the Execute Connector, and it is then possible in a test to execute a command line option (on the host system) using this Connector. The action is not available until you define a Connection with this type.

The command is executed as it would be from the local system command line (so Windows rules on a Windows host, and Linux rules on a Linux system). The available options are Assert that the results contains (using Regular Expressions if desired) or Assert it does NOT contain.

Q) How do I execute a command on Windows?

A) If the command is a Windows executable, you simply call it using the full path. e.g. C:\Windows\System32\write.

If the command is a batch file, or if the command is a command processor (cmd.exe) built-in command such as dir, copy, etc., you need to use the following syntax: "cmd /c <command>" e.g. cmd /c dir c:\

Q) How do I execute a command on Linux or Mac?

A) For Linux and Mac, you enter the command exactly as you would at the command-line.

Q) Why do some of my command line scripting not work? And what can I do about it?

A) Although the first FAQ for Execute Connector says "the command is executed as it would be from the local system command line", that is only 90% accurate. The back end engine runs in a JVM, meaning, like all Java, the engine is in it's own machine and OS (JVM truly means Java Virtual *Machine*). Java has limited access to the host OS. The bottom line is normal shell command-line stuff like piping(), redirection(>), spaces in names, and memory between command don't work.

If your command doesn't work,

1. Make sure the command works on the command line sans Validator.
2. Make sure the command doesn't need a specific current directory.
3. Put the command results in a variable, and read it for any error message.
4. Above all, put the command in a shell or batch file. Then it should behave exactly like the command line because it is loading the shell.

HTTP Connector

Q) What is this connector for?

A) You can use it to test REST or SOAP services. It is a generic connector where you can specify a URL to send data (query string, Form data, or non-form Data) and get a response. You can then compare the response with any string that would indicate success.

Q) I can't connect using SSL

A) SSL isn't currently supported

Q) What type of authentication is available?

A) If you provide values for the user and password when setting up the connection, a Basic Auth header will be sent to the server.

Q) Does it provide a listener to receive data from the Subscriber channel of the SOAP driver?

A) Not at this time, but it is a planned enhancement.

Q) Will the connector work with RBPM REST calls?

A) Not currently. The Authentication model used by RBPM REST services has not been implemented yet.

Q) What is the URL suffix field for?

A) The value of this field is appended to the URL specified in the connection. With it you can add whatever URL suffix is required for the specific test, including a query string.

Q) There are two fields to add data to the http request - Header/Form Data and non-form Data. When would I use one or the other?

A) The REST service will be looking for data on the URL, query string, in form data, or non-form data. Having all three fields available to use when appropriate should allow you to talk to most web services.

Q) How do I specify header data?

A) In the Values field, specify a key/value pair in the form: **h:**field_name=field_value. e.g. h:SOAPAction="\\" will insert into the header (because of the **h:** prefix) the following: SOAPAction=\\". Note that if you need to specify quotes, the quote must be escaped using a backslash.

Q) How do I specify form data?

A) In the Values field, specify a key/value pair in the form: **f:**field_name=field_value. Because of the **f:** prefix, the connector will simulate filling out a form and posting the data.

Q) Can my HTTP call use all three; header data, form data, and normal payload data (in the Data field)?

A) Form data and payload data are mutually exclusive and cannot be used together. If both are included in an action, the form data will be ignored.

Q) Has this connector been tested with SOAP?

A) Typically SOAP is just a matter of sending the right SOAP document as data. Authentication can be tricky and may not be possible in this connector depending on the SOAP service you are using. But give it a try. Paste your SOAP doc in the Data field.

Q) In the Expected Results field, do I have to specify the expected entire return document in order to pass a test?

A) No. In fact if you leave it blank the action will always succeed (since it always matches nothing). You can specify any string in this field that may be returned from the server and the action will succeed. If you check the "Use RegEx" box, you can enter a regular expression in the Expected Results tab.

Q) Can I start a UserApp workflow using the HTTP connector?

A) You can, but you will want to try the User App connector first to be sure it will fit your needs. If you need to do it manually, here is some information that will help.

User App Starting a Workflow via SOAP

To start and approve workflows use the User Application connector. However you can also do this with the HTTP connector as shown below.

1) Add an HTTP connector with the following parameters:

a) http://172.17.2.91:8180/IDMProv/provisioning/service?wsdl (obviously pointing to your server)

b) For a user put a UserAPP provisioning admin and password (same type of user you would use when starting a workflow from a driver)

2) Create a new test with the action Assert/POST result Not Contains (because we'll check for a 500 error if it fails)

3) Enter this SOAP envelope into the data section

You'll need to change the DN of the workflow in the SOAP Envelope and also any needed data items.

i.e.:

Workflow DN: CN=blah123,CN=RequestDefs,CN=AppConfig,CN=User Application Driver,CN=driverset1,O=system

Dataltem Reason = blah

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'>
  <SOAP-ENV:Body>
    <ns1:startRequest xmlns='http://www.novell.com/provisioning/service'
      xmlns:ns1='http://www.novell.com/provisioning/service'>
      <arg0>CN=blah123,CN=RequestDefs,CN=AppConfig,CN=User Application Driver,CN=driverset1,O=system</arg0>
      <arg1>cn=bfox2,ou=users,o=data</arg1>
      <arg2>
        <dataitem>
          <name>reason</name>
          <value>
            <string>blah</string>
          </value>
        </dataitem>
      </arg2>
    </ns1:startRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4) Add this header

SOAPAction = http://www.novell.com/provisioning/service/start

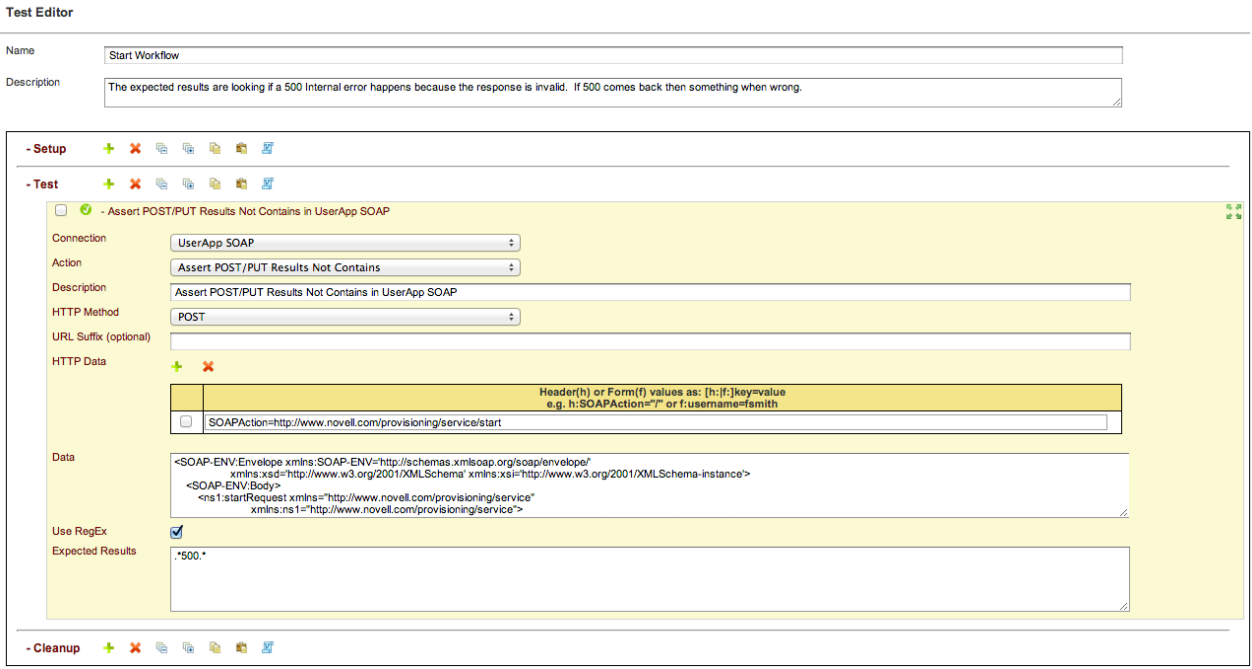
5) Check regex for Expected results and enter this into the expected results ".*500.*" (You can enhance this if you'd like but it's probably sufficient.)

6) Now you can run the test and it will start the workflow.

Optional: If you want to use the InitiatorOverrideDN then the SOAP looks like this. Where arg5 = the InitiatorOverride:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'>
  <SOAP-ENV:Body>
    <ns1:startWithCorrelationIdRequest xmlns='http://www.novell.com/provisioning/service'
      xmlns:ns1='http://www.novell.com/provisioning/service'>
      <arg0>CN=blah123,CN=RequestDefs,CN=AppConfig,CN=User Application Driver,CN=driverset1,O=system</arg0>
      <arg1>cn=bfox2,ou=users,o=data</arg1>
      <arg2>
        <dataitem>
          <name>reason</name>
          <value>
```

```
<string>blah</string>
</value>
</dataitem>
</arg2>
<arg5>cn=ablake,ou=users,o=data</arg5>
</ns1:startWithCorrelationIdRequest>
</SOAP-ENV:Body>
```



```
</SOAP-ENV:Envelope>
```

JDBC Connector

- Q) How does the “Get Column Value and Set Variable” action work?
- A) You must use a SELECT statement in the SQL Statement field that returns a single column. Validator will use the value from the first row returned. If your SELECT statement returns multiple columns and/or rows, Validator will use the value in the first column of the first row.
- Q) How do I configure the JDBC Connector to work with my Oracle Database?
- A) The following steps will work with any JDBC ‘thin’ client - on the Builder tab:
- 1. Copy the correct jdbc thin client jar file to the lib/ext directory. If it has a .zip extension, rename it to .jar.
 - 2. Create a new Connector by clicking the Add (plus sign) icon in the Connections Group
 - 3. Select ‘JDBC Connector’ from the Connector Type drop-down
 - 4. In the Connection Editor Section:
 - a. Provide a Display Name for the Connector
 - b. In the ‘Proxy User’ and ‘Proxy Pwd’ fields, provide the user name and password of the database account to be used for authentication
 - c. In the Driver field, provide the class name of the JDBC driver (Ex: ‘oracle.jdbc.driver.OracleDriver’)
 - d. In the Url field, provide the jdbc connection string (Ex: ‘jdbc:oracle:thin:@172.17.2.146:1521:ORCL10G’)
 - 5. In the Connections Group, check the checkbox to the left of the new connection, and click the Test (play button) icon to test the new connection

- Q) Where should the JAR file be placed when using a JDBC connector?
- A) Place the JAR in the lib/ext directory. If it has a .zip extension, rename it to .jar.

List of common connectors:

SQL Server (net.sourceforge.jtds.jdbc.Driver) - <http://jtds.sourceforge.net/>
Sample URI: jdbc:jtds:sqlserver://172.17.2.150:1470/SampleDatabase

SQL Server (sqljdbc4.jar) - <http://msdn.microsoft.com/en-us/sqlserver/aa937724>
Sample URI: jdbc:sqlserver://172.17.2.150:1470;databaseName=SampleDatabase;

See Known Issues for other issues related to the JDBC Connector.

Text File Connector

- Q) Can it access files on another server other than the Validator server?
- A) It can access any resource mapped or mounted on the server.
- Q) Why are there two ways to access files; via a known filename and via a file pattern?
- A) Oftentimes you won’t know the filename in a directory, like if it has a timestamp in the filename for example. In this case you don’t specify a filename but instead specify a filename pattern (a regex expression) that will match the files you need to access.
- TIP: Hover over the fields when you setup a TextFile connection and it will give you more information.

User App Connector

- Q) When attempting to approve a workflow, I get the error:
- java.lang.RuntimeException: unable to read exception information from stream while executing action method approveRequest
- A) This error is returned by the UserApp API. It is typically returned when you are specifying a piece of data for the approval form that is incorrect, or you don’t specify a piece of data that the approval form requires. It can also be returned if the workflow DN is not correct.

To get a more informative error, you need to do the following:

1) Login to UserApp as the user app admin

2) Administration -> Logging -> Change log level of all above logs (set to Debug)

3) Submit

Then run your test again and look in the User App log file:

IdentityManager/rbpm/jboss/server/IDMProv/log/server.log

Look for something like:

APPROVE failed com.novell.soa.af.impl.soap.AdminException={_Reason=Data item **[reason]** is not defined for activity or provisioning request}

In this case, an attribute of “reason” was included in the Approve Request action in Validator, but “reason” was not on the approval form.

Q) I have a field in the approval form, but if I specify that field name as an “attribute” in the Approve Request action in Validator, I receive the error specified above.

A) Try specifying a default value for the field in the approval activity in the workflow as follows:

1) In Designer, click on the approval activity in the workflow

2) Click the Data Item Mapping tab -> Pre Activity

3) Specify two single quotes (apostrophes) for the Source Expression of the Target Form Field that is causing the error. This gives the field an initial empty value which allows the User App api to recognize the field.

Q) If I start a workflow and immediately try to approve it, I sometimes get an error:

No task for this process/user while executing action method approveRequest.

A) You may need to insert a Wait action so it will sleep for a few seconds after starting the workflow so it will become available to approve.

Sample Error Messages:

General Connector errors:

Bad DN in a FDN field (note the plus sign):

07/20/2011 13:28:36 INFO: ADConnector: assertObjectNotExists() - targetFdn: CN=User,OU+Users,OU=acme,DC=acme,DC=com

07/20/2011 13:28:36 WARNING: LDAPException: Invalid DN Syntax (34) Invalid DN Syntax

LDAPException: Server Message: 0000208F: LdapErr: DSID-0C090654, comment: Error processing name, data 0, vece?

LDAPException: Matched DN:

07/20/2011 13:28:36 INFO: junit.framework.AssertionFailedError: LDAPException: Invalid DN Syntax (34) Invalid DN Syntax

LDAPException: Server Message: 0000208F: LdapErr: DSID-0C090654, comment: Error processing name, data 0, vece?

LDAPException: Matched DN: while executing action method assertObjectNotExists

Typo in attribute name during object creation:

Note the absence of the attribute name. An add event is an atomic operation.

java.lang.AssertionError: LDAPException: Undefined Attribute Type (17) Undefined Attribute Type LDAPException: Matched DN: while executing action method createObject

Typo in attribute name in a modify event:

Note the attribute is named in the error message.

07/19/2011 17:41:03 INFO: EDirConnector: replaceAttributesValues() - attributesValues: {company=[acme widgets]}

07/19/2011 17:41:03 INFO: java.lang.AssertionError: LDAPException: Undefined Attribute Type (17) Undefined Attribute Type LDAPException: Matched DN: while executing action method replaceAttributesValues

eDirectory Connector Specific:

Attribute in object class not yet on user:

Do not forget to specify the object class for the object during creation, and any auxiliary classes as needed modifications.

java.lang.AssertionError: LDAPException: Object Class Violation (65) Object Class Violation LDAPException: Server Message: NDS error: object class violation (-628) LDAPException: Matched DN: while executing action method createObject

Single valued attribute in eDirectory, trying to add a second value:

07/19/2011 17:53:29 INFO: EDirConnector: addAttributesValues() - targetFdn: cn=GeoffTest,ou=Users,o=idv

07/19/2011 17:53:29 INFO: java.lang.AssertionError: LDAPException: Object Class Violation (65) Object Class Violation LDAPException: Server Message: NDS error: illegal attribute (-608)

Previous Change Logs

1.2

Bugs

- Fixed variable expansion to include retry and interval fields in the assert actions.
- If LDAP values are base64 encoded by the ldap server, asserts would fail.
- Fixed LDAP action: Assert Attributes and Values Not Contains where it could result in a false positive.
- Properly handle DN's with escaped commas in browsing and all LDAP actions
- Enlarged Pause and Echo fields to allow easier editing of longer messages. Also allows HTML in message.
- When templates had a Run Cleanup Actions it would not return to original test.
- When a template encountered an error the calling test would still report a success.
- When a Pause or Set Var was in the cleanup section being executed by Run Cleanup Actions, it would not return to the original test.
- Added Description field to Run Templates action
- LDAP browser displays more than 1000 objects. When a limit occurs, it displays the number of object it has gathered.
- Fixed various bugs

Enhancements

- Added variable groups to accommodate multiple test environments
- Added ability to create encrypted variables to use as passwords for different environments
- Added ldap browser to the manage variables page
- Added Base64Encode and Base64Decode to the Generic Action Connection
- Generic Action Asset Variable Contains|Equals will base64 decode value if there is not a match so it can accommodate variables set from LDAP values that are base64encoded.
- If variable values have leading or trailing spaces it is likely user error, so a message is displayed when a test is run to alert the user of the condition.
- Added another Object Import Link to use the Object FDN field as the object to import.
- Ability to import and export tests along with their corresponding variables and connections.
- Enhanced Test/Template Interface to allow collapsing of each type, and drag-and-drop between tests and template lists.
- When editing tests, can drag-and-drop actions between categories.
- License changed to version 2.
- License errors are displayed in browser.
- Added ldap filter to the ldap browser dialog
- Added right-click menu from any action panel within the action editor
- Added right-click menu on the manage variables page
- Added retries to the applicable Execute connector actions

1.1

Bugs

- Fixed parsing issue in json when actions like createobject would not have any attributes defined.
- Fixed JDBC issue where linefeeds would cause JDBC execution issues
- When changing actions, it will preserve common data instead of clearing it.
- When changing connections that have common actions, it will preserve common data.
- Fixed Select All checkbox for connections and tests.
- When pasting actions, any checked tests would be unchecked
- When a test fails, the message in the result log will print the test name instead of the test guid
- When reordering connections or tests, all connections or tests would be deleted.
- After reordering connections, the test suite file could not be saved
- After reordering connections, the Generic Actions connector would be removed
- Fixed issues with reordering conn/tests then deleting one afterward.
- Testing connections would hide all content in the result log after the password line
- Fixed test suite display name issue in Chrome

Enhancements

- Implemented Variables. Can define variables and use them in the tests.
 - Implemented variable browser popup to insert a variable into any text field.
 - Enhanced JDBC connector to allow for multiple sql statements to be executed (separated by ;)
 - Allow user to get back to the results panel (the panel with the progress bar)
 - Implemented new "Classes" field in createObject for LDAP connectors. This allows you to specify classes for the object instead of including them as values for the "objectclass" attribute. This new field is not required so all existing tests continue to run as-is.
 - Implemented schema search buttons for classes and attributes in LDAP connectors.
- In the Create Object action, attributes are limited to the classes specified in the Classes fields. In all other actions, all attributes are displayed.
- Implemented dynamic variables where specific actions will populate a variable. Examples of these follow:
 - Added new LDAP actions:
 - Get Value and Set Variable will retrieve a value from an attribute (if multi-value, get the first value only) and place the value into a variable.
 - Get Object FDN and Set Variable will find the DN of an object from a set of attributes and values and place the DN into a variable.
 - Assert User Login - to test user authentication
 - Assert User Not Login - opposite of above. This is useful for locking an account.
 - Added new JDBC action:
 - Get Value and Set Variable will retrieve a value from a sql statement (get first column from first row returned) and place the value into a variable.
 - Added new Generic action:
 - Set Variable to set a variable to any value
 - Assert Variable Contains to compare a variable to any value
 - Assert Variable Not Contains (opposite of above)
 - Run Cleanup Actions which will execute all cleanup actions before continuing with the test
 - Import an example object in the Create Object action in LDAP based connectors
 - LDAP Tree Browser to find objects in an LDAP tree.
 - Tooltip help for each action. Hover over the action description at the top of each action block.
 - Encrypt passwords in test suite files
 - Added Replace Attribute Value in LDAPConnector to replace a specific value in an attribute.
 - Added border around editable fields in tables when a search icon is present
 - Added placeholder text and borders for editable fields in tables
 - New actions or pasted actions can be inserted anywhere within the category. When adding a new action with the green (+), or when pasting actions, the new action(s) will be inserted after the first checked action, or at the end if no actions are checked.
 - Confirm dialog for deletion of connections, tests, and actions
 - Append the connection name to default action descriptions. If the description has been explicitly changed, it will not modify it.
 - Print current attribute values in the log if none match
 - Added a button to auto-refresh the result log and position the cursor at the bottom
 - When navigating away from Validator, the browser will display a warning about possible unsaved data
 - Prompt to save a modified Test Suite when opening a new file or when creating a new file
 - Removed Retry Count and Retry Interval from properties file since those values were never used.

1.0

- Changed name to Validator for Identity Manager
- Implemented Licensing. 30 demo day license included with install.

0.5

- Fixed issue where actions could not be pasted into empty tests
- Fixed: 706861 - Attribute name seems to be case sensitive in some cases (AD asserts for attributes and values)
- Fixed 709677 - Will not allow the test button to be pressed again while a test is already running
- Fixed issues in HttpConnector. Ability to add header values in both get and put.
- User Interface changes to fit corporate standard
- Improve scrolling so header is always visible
- Fix issue in LDAP connector when adding aux classes with mandatory attributes

0.4

- Ability to expand or collapse all actions within a category. You can also do this on an individual category or action by clicking on the + or - signs. We also fixed an issue where the expanded state was not being saved.
- Added HTTP Connector
- Added Text File Connector
- Ability to disable/enable individual actions
- Reorder Tests and Connections by dragging them to another spot in the list

- Ability to copy Connections and Tests
- Ability to copy and paste multiple Actions - even between different categories or tests
- Generic actions such as comment, echo, pause, wait - have been moved from each connector to their own special connection called: Generic Actions
- LDAP connector features:
 - Added support for SSL
 - Assert Password Not Equals (to trigger Intruder Lockout)
 - Rename and move objects
 - Use wildcards in the CN value for Delete Objects and Assert Object [Not] Exists. For example: cn=testuser*,ou=myou,o=myo would match all users whose CN starts with "testuser". Only * is allowed as a wildcard..
 - Attribute Value Contains allows regular expressions.
 - Can specify multiple attributes and values in Replace action
 - Can compare multiple attributes and values in Assert Attribute Values Contains

Validator for Identity Manager - Architecture

For the technically curious -

The Validator server is an embedded Tomcat server and uses Apache Jersey. It is started via scripts from the command line. Once started, the default URL to access Validator is: <http://localhost:1077/validator>

The client app is HTML and Javascript using JQuery. It communicates with the Validator server via REST calls.

Published by [Google Drive](#) – [Report Abuse](#) – Updated automatically every 5 minutes